



TUGAS AKHIR - TE141599

**PERANCANGAN SUARA TERARAH MENGGUNAKAN
PARAMETRIC ARRAY ULTRASONIK**

Domi Lavaró
NRP 2212100116

Dosen Pembimbing
Dr. Muhammad Rivai, S.T., M.T.
Suwito, S.T., M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE141599

**DESIGNS OF DIRECTIONAL SPEAKER USING
ULTRASOUND PARAMETRIC ARRAY**

Domi Lavaró
NRP 2212100116

Advisors
Dr. Muhammad Rivai, S.T., M.T.
Suwito, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Perancangan Suara Terarah Menggunakan *Parametric Array* Ultrasonik**” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 17 Januari 2017

Domi Lavaró

NRP 2212100116

**Perancangan Saara Terarah Menggunakan
Parametric Array Ultrasonik**

TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Muhammad Rivai, S.T., M.T.
NIP. 196904261994031003

Suwito, S.T., M.T.
NIP. 198101052005011004



PERANCANGAN SUARA TERARAH MENGGUNAKAN PARAMETRIC ARRAY ULTRASONIK

Nama : Domi Lavaró
Pembimbing I : Dr. Muhammad Rivai, S.T., M.T.
Pembimbing II : Suwito, S.T., M.T.

ABSTRAK

Mengarahkan dan memfokuskan gelombang suara belum menjadi hal yang lumrah. Padahal manfaat aplikasi dari sistem audio terarah sangat beragam, salah satunya informasi terfokus dalam suatu ruang bagi seseorang yang melintasi *spot* tertentu, ataupun manfaat yang lain.

Salah satu metode pengarahan suara adalah dengan memodulasi suara audio ke gelombang berfrekuensi ultrasonik yang memiliki keterarahan yang baik. Gelombang ultrasonik juga dapat dibelokkan dengan mengatur beda fase antar elemennya. Pada penelitian ini, digabungkan antara metode pemfokusan suara ultrasonik termodulasi melalui *parametric array* dan metode pengarahan suara *phase-shift array*.

Dalam implementasinya, sinyal audio diinputkan ke *pre-amp* untuk diumpankan ke ADC Arduino. Hasil pembacaan ADC digunakan untuk memvariasi *duty cycle* sinyal PWM 40 kHz yang juga dihasilkan oleh Arduino. Sinyal PWM termodulasi kemudian diteruskan ke serangkaian penguat kelas D untuk kemudian diteruskan ke serangkaian *parametric array* untuk memancarkan gelombang suara.

Hasil menunjukkan frekuensi suara audio dapat dimodulasi secara PWM ke frekuensi ultrasonik untuk lalu dipancarkan dan mengalami demodulasi kembali ke frekuensi audio. Fokus *main lobe* memiliki lebar sudut sebesar 30° dengan level suara sekitar 10 dB nyaris tanpa *side lobe*. Kualitas reproduksi suara relatif baik meskipun dengan *noise*. Sementara itu, terdapat batas sudut pengarahan yang sempit sehingga kurang efektif. Hal ini sebagai efek kompromi rancangan antara fokus dan keterarahan. Diharapkan lanjutan dari penelitian ini dapat menyeimbangkan rancangan keterarahan dan pengarahan gelombang suara.

Kata kunci: Audio terarah; *parametric array*; *phase-shift array*; ultrasonik.

Halaman ini sengaja dikosongkan

DESIGNS OF DIRECTIONAL SPEAKER USING ULTRASOUND PARAMETRIC ARRAY

Name : Domi Lavaró
Advisor I : Dr. Muhammad Rivai, S.T., M.T.
Advisor II : Suwito, S.T., M.T.

ABSTRACT

Directing and focusing sound waves is yet a common thing. Whereas there are several usage of a directional sound, one of those being a focused information source in a room for a particular people that passed a certain spot, among other usage and benefits.

One method to focusing a sound waves is by modulating an audio sound into ultrasound waves which has a good beam focus and a non-linear characteristic. Ultrasound beam could also be redirected by adjusting phase differences between their elements. This study wish to try to combine beam focusing method of using modulated ultrasound by parametric array and beam steering method of using phase-shift array.

For the implementation, audio signal is passed into a pre amp before being fed into Arduino. ADC converts the signal and used its value to vary the duty cycle of a 40 kHz signal which also generated by Arduino. Modulated PWM signal then was fed into a series of class-D amplifier before being passed into parametric array for beaming the sound waves.

The results show that audio frequency could be modulated using PWM into ultrasound frequency for later being beamed and undergo a demodulation back into audio frequency. Main lobe focus covers an area of about 30° with sound level of about 10 dB nearly without side lobes. The sound reproduction quality itself is relatively good with addition of noises. Meanwhile, there is a sound directivity angle limitation which is relatively narrow thus being not effective and couldn't be perceived by listener. This was the effect of design tradeoff between sound focus and directivity. It is expected that any continuation of this study will be able to develop a design that balances beam focus and directivity.

Keywords: Directional sound; parametric array; phase-shift array; ultrasound.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas rahmat, taufik, dan hidayah-Nya, sehingga tugas akhir ini dapat terselesaikan dengan baik. Melalui kegiatan ini, penulis dapat menyelesaikan tugas akhir yang bersifat perancangan perangkat selama satu semester sebagai pengaplikasian ilmu yang telah didapat selama delapan semester.

Buku tugas akhir ini dapat terselesaikan atas bantuan banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibunda Mira Ratnawati dan ayahanda Roval.
2. Bapak Dr. Muhammad Rivai, S.T., M.T. dan Bapak Suwito, S.T., M.T. selaku dosen pembimbing tugas akhir.
3. Bapak Dr. Ir. Hendra Kusuma, M.Eng.Sc., Bapak Ir. Tasripan, M.T., Bapak Ir. Harris Pirngadi, M.T., dan Bapak Fajar Budiman, S.T., M.Sc. selaku dosen penguji.
4. Seluruh Dosen Jurusan Teknik Elektro ITS Surabaya.
5. Keluarga besar Awal Shaleh dan keluarga besar Syaf Effendi beserta rekan dan kerabat orang tua, inter alia Oom Sulistyanto Soejoso, Tante Anne Chadijah, Oom Djoko Wijono (alm.), Oom Kristanto, Bpk. Asri Nugroho, Tante Lusi, Oom Anto, Oom Ed, Tante Sophie, dan Bpk. Iskandar Solichin.
7. Rizky Gigih, Mirza Ferizki, rekan-rekan asisten Laboratorium Elektronika Dasar dan Laboratorium Elektronika Industri, teman-teman Apathetic, teman-teman Vhy, serta teman-teman e52 yang telah memberikan semangat serta bantuan selama berkuliah hingga penyelesaian tugas akhir ini.

Besar harapan penulis bahwa buku tugas akhir ini dapat memberikan informasi dan manfaat bagi mahasiswa Teknik Elektro ITS pada khususnya dan seluruh pembaca pada umumnya.

Surabaya, 17 Januari 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Halaman Judul	i
Pernyataan Keaslian	v
Lembar Pengesahan	vii
Abstrak	i
Abstract	iii
Kata Pengantar	iii
Daftar Isi	vii
Daftar Gambar	ix
Daftar Tabel	xi
Bab I: Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Metodologi	2
1.5 Sistematika Penulisan	3
1.6 Relevansi	4
Bab II: Dasar Teori	5
2.1 Keterarahan Gelombang Suara	5
2.2 Interferensi gelombang	5
2.3 Transduser Ultrasonik	7
2.4 <i>Parametric Array</i>	8
2.5 <i>Phase-Shift Array</i>	9
2.6 <i>Pulse Width Modulation</i>	11
2.7 Amplifier kelas D	12
2.8 Arduino Due	12
Bab III: Perancangan Sistem	15
3.1 Konsep Rancangan	15
3.1.1 Rancangan <i>Hardware</i>	15
3.1.2 Rancangan <i>Software</i>	16
3.2 Subsistem	18
3.2.1 <i>Pre-amp</i>	18
3.2.2 Arduino	19
3.2.3 <i>Time delay</i>	22
3.2.4 Amplifier	24
3.2.5 <i>Parametric Array</i>	25
Bab IV: Pengujian dan Analisis	27

4.1 Analisis elektrik.....	27
4.2 Analisis akustik	31
Bab V: Penutup.....	35
5.1 Kesimpulan.....	35
5.2 Saran.....	35
Daftar Pustaka.....	37
Lampiran.....	39
Biodata Penulis	49

DAFTAR GAMBAR

Gambar 2.1	Visualisasi hasil interferensi dua gelombang.....	6
Gambar 2.2	Gelombang baru akibat interferensi dua gelombang.	6
Gambar 2.3	Transduser piezoelektrik untuk ultrasonik.....	7
Gambar 2.4	Contoh susunan <i>parametric array</i>	8
Gambar 2.5	Visualisasi arah gelombang pada <i>phase-shift array</i>	9
Gambar 2.6	Visualisasi PWM dengan variasi <i>duty cycle</i>	11
Gambar 2.7	Rangkaian dasar <i>amplifier</i> kelas D	12
Gambar 3.1	Blok diagram struktur <i>hardware</i>	15
Gambar 3.2	Visualisasi proses modulasi PWM pada Arduino	16
Gambar 3.3	Grafik <i>flowchart</i> algoritma <i>software</i>	17
Gambar 3.4	Skematik <i>preamp</i>	18
Gambar 3.5	Realisasi rangkaian <i>preamp</i>	19
Gambar 3.6	<i>Timing diagram</i> proses modulasi PWM	21
Gambar 3.7	Simulasi <i>beam pattern</i> tanpa <i>delay</i>	23
Gambar 3.8	Simulasi <i>beam pattern</i> dengan <i>delay</i>	23
Gambar 3.9	Skematik <i>amplifier</i> kelas D.....	24
Gambar 3.10	Realisasi rangkaian <i>amplifier</i>	25
Gambar 3.11	Skematik rangkaian <i>parametric array</i>	26
Gambar 3.12	Realisasi <i>parametric array</i>	26
Gambar 4.1	Sinyal audio 440 Hz.....	27
Gambar 4.2	Sinyal output <i>pre amp</i>	28
Gambar 4.3	Sinyal Arduino PWM 40 kHz, <i>duty cycle</i> 50%	28
Gambar 4.4	Sinyal Arduino PWM termodulasi, <i>duty cycle</i> dinamis ...	29
Gambar 4.5	Dua sinyal PWM dengan <i>delay</i>	29
Gambar 4.6	Sinyal output <i>opto-coupler</i> H11L1	30
Gambar 4.7	Sinyal output <i>amplifier</i> kelas D	30
Gambar 4.8	Kegiatan pengukuran	31

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Koneksi pin aktif pada Arduino Due	21
Tabel 4.1 Pengukuran level suara tanpa <i>delay</i> pada jarak 30 cm.	32
Tabel 4.2 Pengukuran level suara tanpa <i>delay</i> pada jarak 1 meter	32
Tabel 4.3 Pengukuran level suara dengan <i>delay</i> 25 μ s	33

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Meski perkembangan berbagai teknologi pada *speaker* cukup pesat, terdapat satu aspek pada *speaker* yang relatif cukup lambat perkembangannya, yaitu aspek kontrol terhadap distribusi dan pemosisian suara. *Speaker* konvensional sangatlah handal dalam menghasilkan suara yang seragam untuk satu lingkup ruangan. Namun, tidak dimungkinkan untuk membuat seseorang mendengarkan satu suara tertentu tanpa gangguan suara lain. Padahal, aplikasi dari sistem audio terarah sangat beragam, seperti misalnya: iklan komersial di ruang terbuka yang ditujukan hanya untuk pejalan kaki yang melintas *spot* tertentu; sistem audio personal dalam suatu ruangan; informasi personal dalam suatu pameran lukisan yang menjelaskan kepada pengunjung mengenai lukisan di depannya; suara navigasi yang hanya dapat didengar oleh pengemudi mobil / bus namun tidak oleh penumpang.

Suara audio dapat diarahkan dan difokuskan pada suatu area tertentu dengan memanfaatkan sifat interferensi dari dua buah gelombang suara dan juga sifat non-linier dari suara berfrekuensi tinggi [1]. Interferensi dua buah gelombang yang merambat sejajar dapat menghasilkan dua frekuensi baru, yaitu senilai jumlah kedua frekuensi tersebut serta senilai selisihnya. Dua buah suara ultrasonik dapat diatur frekuensinya untuk menghasilkan frekuensi “selisih” yang dapat didengar (*audible*) sementara frekuensi “jumlah” diabaikan. Suara ultrasonik dipilih sebagai medium *carrier* karena dengan panjang gelombang yang sangat pendek, memiliki fokus keterarahan yang cukup tinggi serta tidak dapat didengar secara normal oleh manusia.

Meskipun begitu, *speaker* terarah memiliki beberapa kesulitan dalam perwujudannya. *Speaker* jenis ini umumnya menghasilkan kualitas suara yang terbilang buruk dan distorsi yang relatif besar [1]. Faktor-faktor penting yang memengaruhi antara lain adalah proses modulasi sumber suara ke frekuensi ultrasonik dan proses amplifikasi.

1.2 Rumusan Masalah

Permasalahan utama yang dibahas dalam tugas ini adalah bagaimana mengarahkan gelombang suara dengan menggunakan implementasi modulasi suara audio ke frekuensi ultrasonik.

1. Bagaimana memodulasi suara audio ke frekuensi ultrasonik?
2. Bagaimana mengarahkan suara tanpa mengubah sudut mekanik *speaker*?

1.3 Tujuan

Tujuan yang diharapkan dari tugas akhir ini adalah sebagai berikut:

1. Implementasi modulasi suara audio dengan teknik PWM ke frekuensi ultrasonik guna memfokuskan arah gelombang akustik.
2. Implementasi *phased array* guna mengarahkan gelombang akustik dengan sudut tertentu.

1.4 Metodologi

Metodologi yang digunakan dalam perancangan tugas akhir ini adalah:

1. Studi literatur
Mengumpulkan dan mempelajari buku serta referensi yang berhubungan dengan pengarah gelombang suara. Hal ini dilakukan untuk menambah pemahaman mengenai konsep permasalahan rancangan yang ingin dicapai dan mengetahui langkah-langkah yang diperlukan dalam merancang sistem yang dikehendaki.
2. Perancangan sistem dan simulasi
Dari hasil pembelajaran literatur, dilakukan iterasi konsep sistem dan desain rancangan beserta pula rangkaian elektronik yang disimulasikan.
3. Realisasi rancangan
Tahap dimana hasil desain direalisasikan ke dalam bentuk fisik berupa sebuah *speaker* yang menghasilkan suara terarah.

4. Uji coba dan analisis data
Proses pengujian perangkat dilakukan dengan pengukuran gelombang suara pada tiap-tiap subsistem dan pengukuran frekuensi suara di daerah sekitar *speaker*, baik frekuensi ultrasonik maupun audiosonik. Dilakukan pula pengujian tanpa pengukuran oleh manusia untuk mendengarkan suara-suara di daerah sekitar *speaker*.
5. Pembuatan laporan tugas akhir
Laporan tugas akhir berisi pembahasan dan analisis dari sistem yang telah dirancang. Disertakan pula kesimpulan yang didapatkan seperti kelebihan dan kekurangan sistem, termasuk saran atau rekomendasi untuk perancangan berikutnya.

1.5 Sistematika Penulisan

Laporan tugas akhir ini akan tersusun atas 5 bab dengan sistematika penulisan sebagai berikut:

Bab 1: PENDAHULUAN

Bab ini meliputi latar belakang beserta rumusan permasalahan, ide rancangan yang ingin dicapai, tujuan, dan manfaat, dari tugas akhir ini, penjabaran mengenai metodologi pengerjaan dan sistematika penulisan.

Bab 2: TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai teori-teori dan kajian pustaka yang digunakan dalam pengerjaan tugas akhir ini.

Bab 3: PERANCANGAN SISTEM

Bab ini menjabarkan mengenai metode yang akan digunakan dalam proses perancangan dan desain sistem *speaker* suara terarah.

Bab 4: HASIL DAN ANALISIS

Bab ini membahas mengenai hasil dan karakteristik dari unit *speaker* yang telah dirancang.

Bab 5: PENUTUP

Bab ini berisi kesimpulan beserta saran – termasuk kelebihan dan kekurangan – dari hasil rancangan.

1.6 Relevansi

Dengan tugas akhir ini diharapkan dapat memberikan manfaat yaitu sebagai salah satu referensi dalam merancang suatu sistem pemancar suara terarah menggunakan media gelombang suara frekuensi ultrasonik.

BAB II

DASAR TEORI

2.1 Keterarahan Gelombang Suara

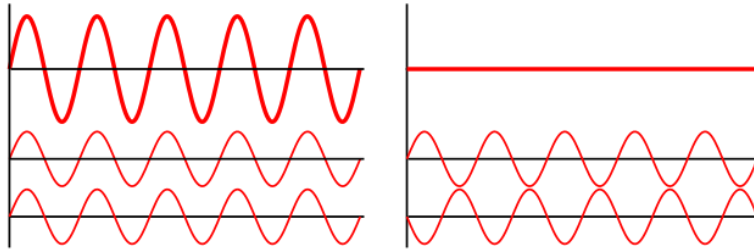
Suatu gelombang umumnya dipahami sebagai energi yang bergerak ke segala arah akibat difraksi. Semakin jauh gelombang merambat, maka semakin besar pula area yang ditempuh oleh gelombang yang menyebar. Kemudian, energi gelombang tersebut akan semakin berkurang hingga akhirnya hilang seutuhnya [2].

Dasar dari pengarahan suara berkaitan erat dengan rasio panjang gelombang terhadap lebar *wavefront* atau ukuran sumber suaranya. Untuk sumber suara yang lebih kecil dibandingkan panjang gelombangnya, suara berpropagasi ke segala arah. Sebaliknya, gelombang suara akan lebih terarah jika panjang gelombang suara lebih pendek dari lebar sumber suaranya [1].

Suara audiosonik umumnya memiliki panjang gelombang antara beberapa sentimeter hingga beberapa meter [3], sehingga *speaker* konvensional umumnya akan menghasilkan suara yang menyebar. Alih-alih, transduser ultrasonik umum yang memancarkan suara ultrasonik dengan panjang gelombang beberapa milimeter – sekitar 8mm pada frekuensi 40 kHz – akan menghasilkan lebar difraksi yang lebih kecil [4].

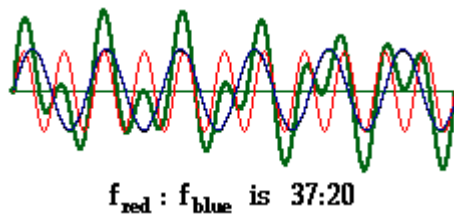
2.2 Interferensi gelombang

Interferensi adalah suatu fenomena dimana dua gelombang mengalami superposisi sehingga menghasilkan gelombang baru yang memiliki amplitudo yang sama, lebih lemah, ataupun lebih kuat. Interferensi umumnya mengacu pada interaksi gelombang yang berhubungan atau koheren satu sama lain, baik dikarenakan gelombang-gelombang tersebut berasal dari sumber yang sama maupun karena memiliki frekuensi yang mirip/berdekatan.



Gambar 2.1 Visualisasi hasil interferensi dua gelombang (kiri) interferensi konstruktif dan (kanan) interferensi destruktif (diambil dari <http://wikimedia.org/>)

Prinsip dasar superposisi gelombang menyatakan bahwa ketika dua atau lebih gelombang dengan jenis yang sama bertemu di suatu titik yang sama, resultan amplitudo di titik tersebut sama dengan penjumlahan vektor dari amplitudo masing-masing gelombang [5]. Interferensi konstruktif terjadi ketika beda fase antar gelombang adalah kelipatan genap dari π (180°), sementara interferensi destruktif terjadi ketika beda fasenya adalah kelipatan ganjil dari π . Jika perbedaan antar fase ada di antaranya, maka magnitudo perpindahan dari gelombang jumlah berada di antara nilai maksimum dan minimum. Ketika terdapat dua gelombang dengan frekuensi yang berbeda, maka hasil interferensinya akan menghasilkan gelombang dengan frekuensi baru. Seperti diilustrasikan oleh Gambar 2.2, superposisi sinyal merah dan sinyal biru menghasilkan sinyal hijau.



Gambar 2.2 Contoh gelombang baru akibat interferensi dua gelombang.

Suara ultrasonik memiliki sifat non-linier ketika merambat melalui suatu medium. Selayaknya sifat non-linier lemah, hal ini dapat diekspresikan dalam rumus deret Taylor seperti berikut [1]:

$$y = \sum b_{i,j}(\sin(\omega_i + \omega_j) + \sin(\omega_i - \omega_j)) \quad (2.1)$$

Dapat terlihat bahwa sifat non-linier dan interferensi dua gelombang sejajar menghasilkan gelombang yang terdiri dari frekuensi penjumlahan dan frekuensi selisih dari dua buah frekuensi asalnya. Jika $\omega_{i,j}$ berada dalam rentang frekuensi ultrasonik, maka $\omega_i + \omega_j$ dapat diacuhkan karena tentu juga akan menghasilkan frekuensi ultrasonik yang lebih tinggi. Namun sebaliknya, $\omega_i - \omega_j$ akan dapat menghasilkan suara yang dapat didengar (*audible*) apabila selisih kedua frekuensi referensi tidak terlalu jauh. Sifat non-linier tersebut menyebabkan transfer energi dari frekuensi ultrasonik menjadi frekuensi yang dapat didengar.

2.3 Transduser Ultrasonik

Transduser ultrasonik yang dipakai adalah jenis transduser piezoelektrik. Hal ini untuk mengakomodasi rencana rancangan yang menggunakan modulasi PWM alih-alih menggunakan modulasi AM yang cocok menggunakan film piezoelektrik. Transduser ultrasonik umumnya dijumpai pada modul sensor pengukuran jarak elektronik.



Gambar 2.3 Transduser piezoelektrik untuk ultrasonik
(diambil dari <http://www.futurlec.com/>)

2.4 Parametric Array

Parametric array menghasilkan suara ultrasonik dengan menggunakan serangkaian transduser piezoelektrik, berbeda dengan *speaker* konvensional yang menggunakan kumparan elektromagnetik tunggal. Transduser piezoelektrik dipilih karena terbuat dari kristal yang mampu bergetar puluhan ribu kali dalam satu detik. Serangkaian transduser kecil dalam jumlah banyak digunakan untuk efek demodulasi gelombang ultrasonik di udara yang menghasilkan suara audio terdengar lebih pelan dibandingkan intensitas ultrasonik. Juga, *parametric array* mengurangi efek difraksi dibandingkan dengan transduser tunggal [1].

Dalam bidang akustik, suatu *parametric array* adalah mekanisme transduksi non-linier yang menghasilkan sorotan gelombang terfokus, nyaris tanpa *side-lobe*. *Parametric array* memanfaatkan interaksi non-linier gelombang frekuensi tinggi untuk membatasi difraksi [6]. Rangkaian elektronik kemudian dihubungkan dengan transduser piezoelektrik secara konstan mengatur frekuensi dari dua buah gelombang agar nantinya dihasilkan frekuensi rendah yang dapat didengar ketika kedua gelombang tersebut melebur dan terdemodulasi. Keseluruhan proses ini disebut sebagai interaksi parametrik [2] [7].

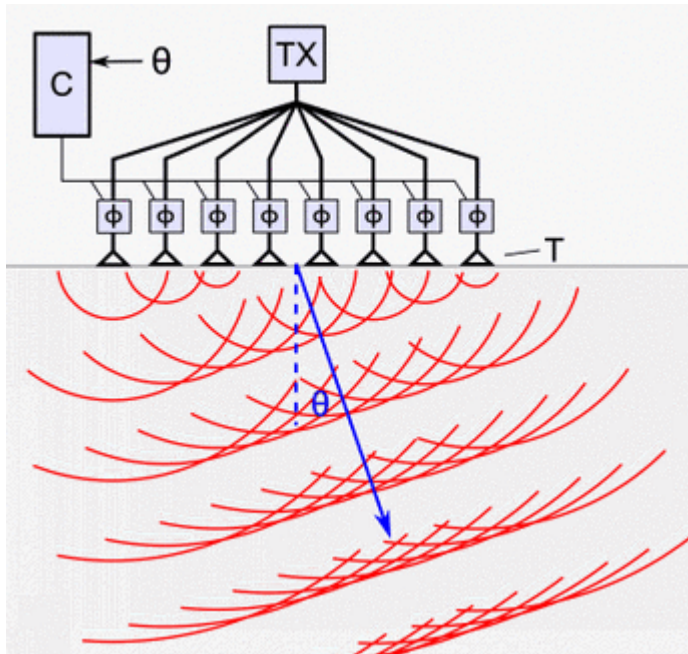


Gambar 2.4 Contoh susunan *parametric array*
(diambil dari <http://dm.ncl.ac.uk/>)

2.5 Phase-Shift Array

Phase-shift array mengacu pada serangkaian/susunan pemancar gelombang (semisal antenna atau transduser ultrasonik) dimana sudut pancaran gelombang yang dihasilkan dapat diarahkan secara elektrik tanpa menggerakkan fisik perangkat [8]. Teknik pembelokan yaitu dengan serangkaian gelombang dengan beda fase sepersekian derajat antar gelombang-gelombang tersebut.

Aplikasi lazim *phased array* ultrasonik adalah sebagai *nondestructive testing* (NDT) antara lain di bidang medis untuk memindai jaringan organ dalam dan di bidang industri untuk mencari celah cacat pada produksi logam. Pada bidang industri, parameter frekuensi yang digunakan dapat mencapai 1 MHz.



Gambar 2.5 Visualisasi arah gelombang pada *phase-shift array*

Mengacu pada Gambar 2.5, rangkaian *phase-shift array* pada dasarnya terdiri dari osilator (TX) yang menghasilkan sebuah sinyal elektronik tersusun atas pulsa-pulsa gelombang sinusoidal terosilasi pada frekuensi ultrasonik, yang mana kemudian diteruskan ke serangkaian (*array*) transduser ultrasonik (T). Waktu pemancaran gelombang (*timing*) oleh setiap transduser diatur oleh unit *programmable delay* (Φ) yang dikontrol oleh sebuah mikroprosesor sistem kontrol (C). Garis-garis lengkung merah adalah rambatan gelombang (*wavefront*), yang mengalami superposisi membentuk gelombang planar/datar, menciptakan pancaran suara yang menempuh ke suatu arah spesifik. Sudut planar (θ) dapat diubah dengan memvariasi waktu *delay* progresif dari setiap transduser ataupun pergeseran fase gelombangnya.

Secara matematis, *phased array* adalah suatu bentuk contoh dari kisi difraksi dimana setiap transduser ultrasonik bertindak serupa seperti celah sempit yang memancarkan gelombang. Selayaknya sebuah kisi difraksi, diasumsikan celah sumber gelombang sejumlah N , masing-masing diposisikan berjarak d menransmisikan gelombang sepanjang λ . Jika jumlah pemancar dan jarak antar tiap-tiap pemancar tersebut adalah tetap, maka dengan mengikutkan variabel beda fase (ϕ) dapat diturunkan rumus untuk menghitung sudut θ , yaitu

$$\theta = \sin^{-1} \left(-\frac{\lambda}{2\pi d} \phi \right) \quad (2.2)$$

Jadi, dapat dilihat bahwa sudut dengan amplitudo maksimum sebagai fungsi dari panjang gelombang dan pergeseran fase gelombang. Selain itu, jika dilakukan substitusi berdasarkan persamaan dasar gelombang bahwa $\lambda \cdot f = v$ serta persamaan hubungan beda fase dan *delay* waktu bahwa $\phi/f = t_d$, maka didapatkan persamaan

$$\theta = \sin^{-1} \left(-\frac{v}{2\pi d} t_d \right) \quad (2.3)$$

dimana v adalah kecepatan suara di udara sebesar 340 m/s. Didapatkan persamaan yang menghubungkan sudut daya maksimum (θ) dengan waktu *delay* (t_d) antar elemen [9]. Persamaan tersebut tidak bergantung pada panjang gelombang, mengindikasikan dapat diterapkan untuk rentang frekuensi yang sangat besar.

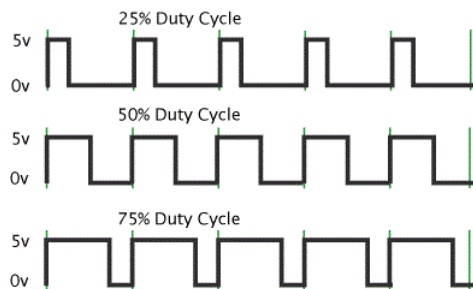
Selain itu, terdapat batasan sudut pembelokan maksimum bagi suatu *phase shift array*. Batas sudut ini bergantung pada rasio panjang gelombang terhadap lebar sumber gelombang ataupun lebar *wavefront* [10]. Perumusannya adalah sebagai berikut:

$$\theta_{max} = \sin^{-1} \left(0.514 \frac{\lambda}{A} \right) \quad (2.4)$$

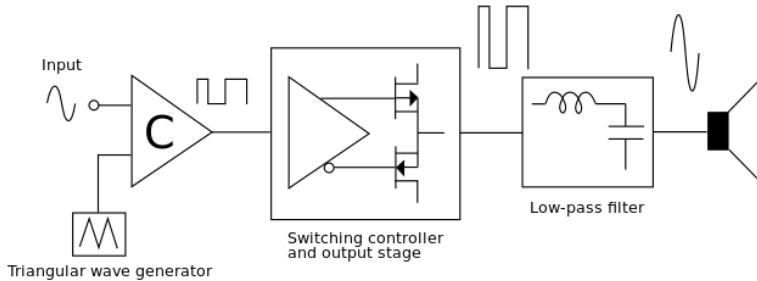
2.6 Pulse Width Modulation

Pulse-width modulation (PWM) adalah suatu teknik modulasi untuk memodulasi sinyal ke bentuk sinyal pulsa, dengan melakukan variasi *duty cycle*. Istilah *duty cycle* mendeskripsikan perbandingan waktu ‘on’ terhadap interval atau periode satu gelombang. *Duty cycle* diekspresikan dalam persentase, dengan 100% berarti sinyal selalu dalam status *on/high*. *Duty cycle* yang rendah identik dengan daya rendah karena sinyal lebih sering berstatus ‘off’.

Modulasi ini dapat digunakan untuk mengatur suplai daya ke perangkat listrik seperti lampu atau motor, yaitu dengan memvariasi rata-rata nilai tegangan antara kondisi *on/off*. Selain itu, PWM dapat pula digunakan sebagai salah satu metode *encoding* informasi analog menjadi digital. Dalam penelitian ini, penggunaan PWM adalah untuk mengoperasikan transduser ultrasonik, antara lain juga terkait dengan faktor kualitas (*q*) dan *ringing*. [11]



Gambar 2.6 Visualisasi PWM dengan variasi *duty cycle*
(diambil dari <http://www.arduino.cc/>)



Gambar 2.7 Rangkaian dasar *amplifier* kelas D
(diambil dari <http://wikimedia.org/>)

2.7 Amplifier kelas D

Amplifier kelas D dirancang berdasarkan prinsip PWM.

Keuntungan *amplifier* kelas D antara lain [12]:

- Rugi daya sebagai disipasi panas relatif minim
- Pengurangan ukuran, biaya, dan berat perangkat *amplifier* karena tidak diperlukan *heat sink* yang besar atau bahkan tidak diperlukan sama sekali.
- Konversi daya dengan efisiensi sangat tinggi, umumnya di atas 90% pada daya maksimum dan sekitar 50% pada daya rendah
- Dapat bekerja dari sumber sinyal digital tanpa terlebih dahulu memerlukan konversi sinyal ke analog menggunakan DAC.

2.8 Arduino Due

Arduino Due merupakan *board* Arduino pertama yang berbasis mikrokontroler ARM 32-bit menggunakan tipe SAM3X. SAM3X memiliki *flash memory* sebesar 512 kB (terdiri dari 2 blok 256 kB) untuk menyimpan *code* pemrograman, sementara S-RAM yang tersedia adalah sebesar 96 kB (terdiri dari sebuah blok 64 kB dan sebuah blok 32 kB). Semua memori yang tersedia (*flash*, RAM, dan ROM) dapat diakses

secara langsung sebagai *flat addressing space* atau menggunakan *direct memory access* (DMA).

Board ini dilengkapi dengan 54 pin *digital input/output* (12 di antaranya dapat digunakan sebagai output PWM), 12 *analog input*, 4 UART (*serial port*), sebuah *clock* 84 MHz, dan lain sebagainya. Namun, perlu diingat bahwa tidak seperti kebanyakan *board* Arduino lain yang bekerja pada tegangan 5V, Arduino Due menggunakan tegangan 3,3V. Menerapkan tegangan lebih dari 3,3V ke pin-pinnya dapat merusak *board* Due.

Atas dasar spesifikasinya, Arduino Due dipilih karena memiliki kecepatan yang lebih tinggi dibandingkan kebanyakan *chip* mikrokontroler lain dan kemampuannya untuk melakukan *multitasking*. Dipadukan dengan ukuran memori yang relatif besar, Arduino Due memiliki potensi untuk melakukan pengolahan sinyal digital (DSP) terkait sinyal audio yang diinputkan. Penggunaan *board* Arduino sendiri bertujuan untuk mempermudah pemrograman dengan menggunakan Arduino IDE maupun *software* Atmel Studio.

Diagram rinci *pin out* Arduino Due terlampir.

Halaman ini sengaja dikosongkan

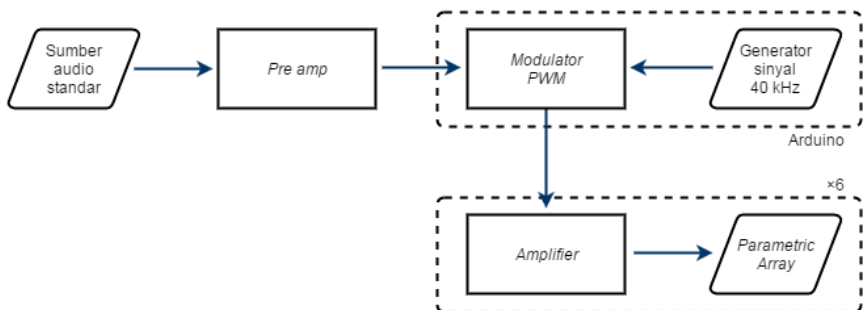
BAB III

PERANCANGAN SISTEM

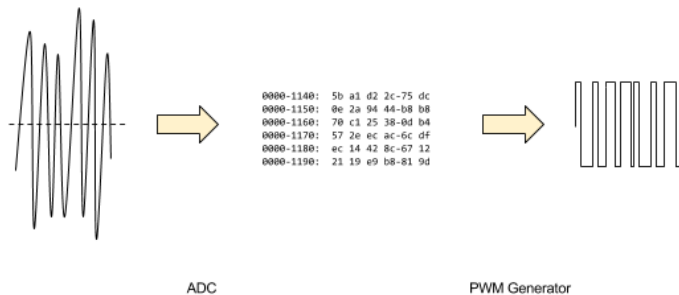
3.1 Konsep Rancangan

3.1.1 Rancangan *Hardware*

Struktur dari desain *hardware* diilustrasikan dalam blok diagram pada Gambar 3.1. Input sinyal berasal dari sumber audio standar. Selanjutnya, untuk dapat menggunakan ADC Arduino secara optimal, diperlukan sinyal yang bervariasi dalam rentang 0–3,3V. Untuk itu, digunakan suatu *preamp* untuk mengatur bias sinyal dan menguatkannya sebelum diumpankan ke Modulator. ADC mengindikasikan nilai pembacaan digital dari input sinyal menuju ke mikrokontroler. Mikrokontroler kemudian melakukan perhitungan untuk mengatur *duty cycle* dari sinyal pulsa 40 kHz yang dihasilkan oleh Arduino. Sinyal PWM dari pin output diumpankan ke masing-masing *amplifier* untuk dikuatkan lebih lanjut, sebelum akhirnya diteruskan ke transduser ultrasonik. Meskipun semua rangkaian elektronik rentan terhadap *noise*, tidak dipasangkan suatu *high-pass filter* sebelum transduser dengan menimbang bahwa transduser hanya mampu menghasilkan gelombang akustik di sekitar frekuensi 40 kHz. Dengan kata lain, transduser secara tidak langsung bertindak sebagai filter.



Gambar 3.1 Blok diagram struktur *hardware*

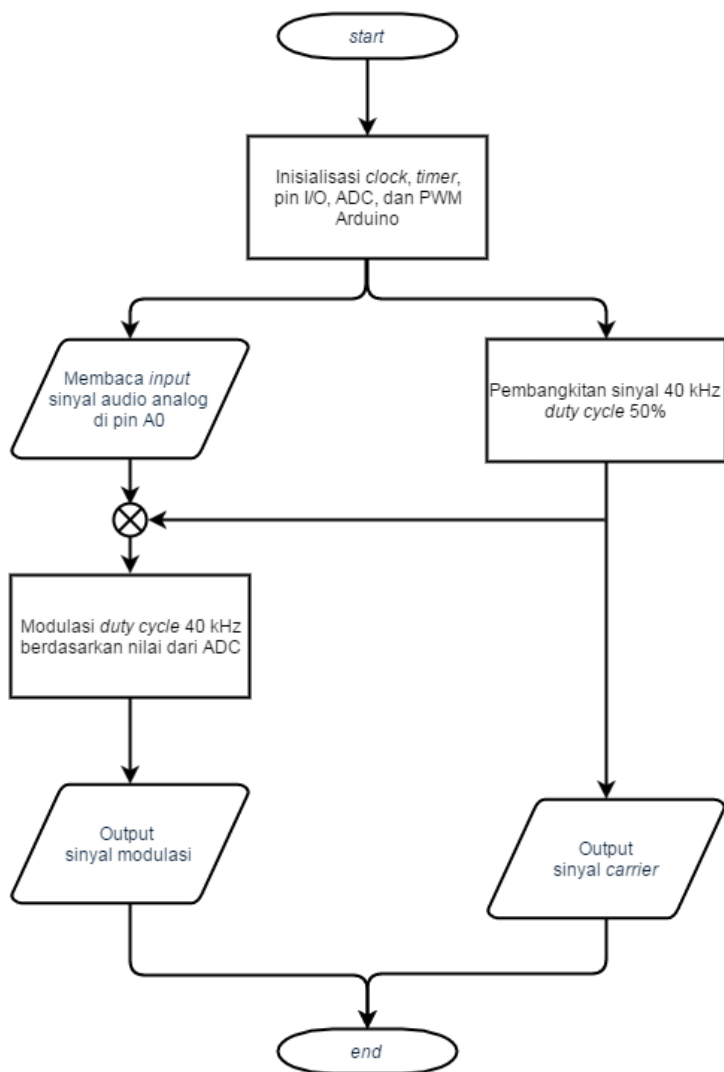


Gambar 3.2 Visualisasi proses modulasi PWM pada Arduino

3.1.2 Rancangan *Software*

Melanjutkan rancangan *hardware* yang telah dijelaskan sebelumnya, fungsi generator gelombang ultrasonik serta proses modulasi dilakukan dalam satu modul menggunakan *board* Arduino. Proses yang dilakukan oleh Arduino sendiri pada dasarnya ada dua, seperti diilustrasikan pada Gambar 3.2, yaitu proses *sampling* sinyal input analog dan proses pembangkitan sinyal persegi 40 kHz dengan variasi *duty cycle*. Adapun proses kerja atau algoritma pemrograman Arduino dijabarkan oleh *flowchart* pada Gambar 3.3.

Saat dinyalakan, Arduino akan melakukan proses inisiasi dan konfigurasi *clock/timer*, ADC, PWM, serta pin-pin yang akan digunakan. Pada umumnya *board* Arduino memiliki keterbatasan akibat dari *default library* yang disediakan oleh Arduino, sehingga proses inisiasi penting untuk bisa memanfaatkan potensi maksimum dari mikrokontroler, misalnya seperti pembagian waktu *timer* selain dengan nilai *prescaler*, menonaktifkan fungsi yang menghambat kerja dan tidak diperlukan, serta mengakses fitur khusus dari suatu mikrokontroler.



Gambar 3.3 Grafik *flowchart* algoritma *software*

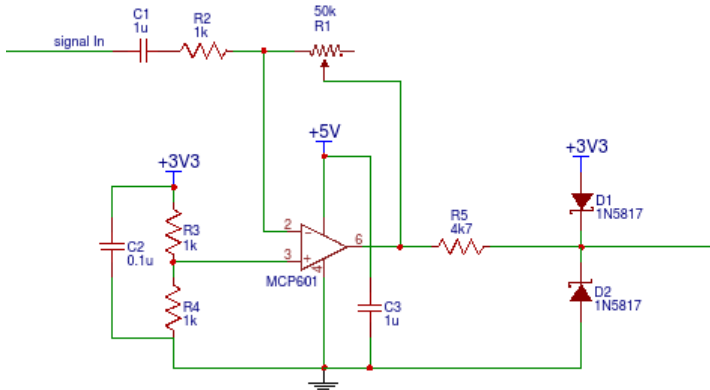
Sesuai fungsinya, ADC melakukan konversi nilai analog ke digital bagi sinyal yang diinputkan. Guna menjaga kualitas sinyal input, resolusi yang digunakan adalah 12 bit sebagai resolusi maksimum pada Arduino Due. Namun, untuk mencocokkan dengan frekuensi output sebesar 40 kHz, maka laju pembacaan ADC adalah sebanyak 40.000 *sampling* per detik.

Sementara itu, generator PWM pada Arduino diinisiasi menghasilkan gelombang 40 kHz dengan *duty cycle* 50%. Hasil dari pembacaan ADC digunakan untuk memvariasi *duty cycle* PWM tiap-tiap satu periode gelombang atau setiap 25 μ s. Secara umum, persentase *duty cycle* berbanding lurus terhadap nilai pembacaan ADC. Selanjutnya, gelombang PWM 40 kHz dikeluarkan melalui pin output digital yang telah ditentukan.

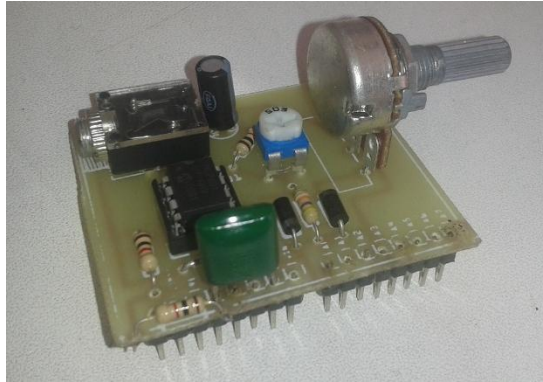
3.2 Subsistem

3.2.1 Pre-amp

Pre-amp dirancang untuk melakukan pergeseran bias dan penguatan awal sinyal input audio dari rentang -1V – 1V menjadi rentang tegangan 0V – 3,3V yang cocok dengan rentang tegangan ADC Arduino Due. Op-amp yang digunakan pada *preamp* ialah tipe MCP601.



Gambar 3.4 Skematik *preamp*



Gambar 3.5 Realisasi rangkaian *preamp*

Tahap awal adalah suatu rangkaian *high-pass filter* yang diimplementasikan dengan sebuah kapasitor dan rangkaian pembagi tegangan. Frekuensi sudut dari HPF input adalah $\frac{1}{RC} = \frac{1}{0.1\mu F \cdot 1k\Omega} = 0.1\text{ kHz} = 100\text{ Hz}$. Nilai rendah ini dirasa cukup sebagai filter *noise* dan melewati frekuensi penting yang diinginkan.

Tahap berikutnya berupa penggeseran nilai bias dan penguatan sinyal input. Rangkaian pembagi tegangan simetris antara 3,3V dan *ground* akan memberikan *bias offset* senilai $\frac{1k}{1k+1k} \cdot 3,3V = 1,65V$. Nilai bias ini cocok untuk rentang 3,3V. Untuk amplifikasi, faktor penguatannya senilai 1,65V. Fungsi trim-pot sendiri adalah untuk menentukan rasio amplifikasi, agar frekuensi rendah tidak serta-merta mencapai *peak* tegangan namun tegangan frekuensi tinggi juga tidak terlalu kecil. Secara bersamaan, maka didapatkan nilai rentang baru antara 0V – 3,3V.

3.2.2 Arduino

Seperti telah dijelaskan sebelumnya, Arduino berfungsi sebagai generator sinyal ultrasonik dan sebagai modulator, dengan proses kerja berupa *sampling* sinyal analog (ADC) dan pembangkitan sinyal 40 kHz dengan variasi *duty cycle* (PWM). Di sini, Arduino diprogram dengan bahasa C menggunakan aplikasi

Atmel Studio 7.0 dengan ekstensi Atmel Software Framework versi 3.28.0. Rincian program terlampir.

Menggunakan *default library* yang disediakan oleh Arduino, fungsi ADC (`analogRead`) dan fungsi PWM (`analogWrite`) memiliki keterbatasan yang mempersulit serta tidak mendukung capaian yang akan dituju. Pada pemakaian standar, perintah `analogRead` memiliki waktu pembacaan sekitar 100 μ s ataupun 40 μ s untuk satu konversi, belum termasuk potensi hambatan oleh proses lain seperti *interrupt*. Pengaturan parameter ADC yang disediakan oleh Arduino dapat memperbaiki kecepatan konversi hingga 1 μ s atau 1 juta *sampling* per detik, namun semua parameter yang ada mengubah kecepatan konversi secara acak dan tidak menentu sehingga tidak dapat diandalkan untuk memperoleh laju konversi tertentu. Sementara itu, perintah `analogWrite` pada pemakaian standar menghasilkan 490 Hz ataupun 980 Hz dengan *duty cycle* statis.

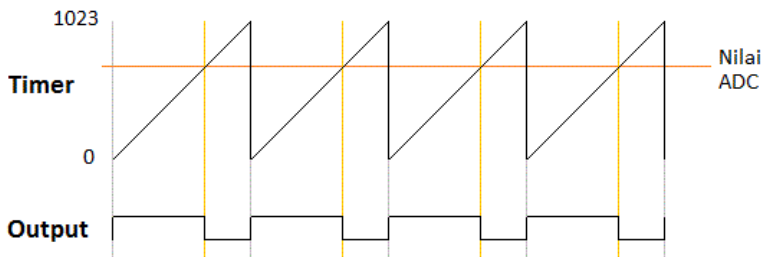
Untuk menghasilkan PWM dengan frekuensi tertentu dan *duty cycle* dinamis juga tidak mudah. Teknik generator PWM yang digunakan bukan berupa metode *bit-banging* namun menggunakan *timer* yang menghitung secara berkala dan berulang-ulang untuk kemudian dihasilkan output *on* setiap *timer* bernilai 0 dan output *off* saat *timer* menyamai nilai input *compare register*. Perlu diperhatikan bahwa teknik PWM berkaitan erat dengan manipulasi *clock* dan *timer* yang berarti bisa saja mengganggu laju kinerja proses lain yang peka terhadap waktu. Dengan mempertimbangkan hal-hal tersebut, untuk itu digunakan *library* yang disediakan Atmel Studio untuk mengakses dan memanipulasi fitur mikroprosesor secara lebih langsung.

Pada permulaan program, dilakukan inisiasi bagi *board* Arduino Due. Dilakukan pengaturan *timer0* untuk menghasilkan *clock* 40 kHz, serta juga pengaturan pembangkitan gelombang 40 kHz dengan *duty cycle* pasif 50% untuk nantinya siap dimodulasi. Adapun koneksi pin yang digunakan dirinci pada tabel 3.1. Definisi pin diperlukan karena adanya *multiplexing* pada pin Digital I/O Arduino dengan pin fungsi SAM3X8E, dalam hal ini *line I/O PWM*.

Tabel 3.1 Koneksi pin aktif pada Arduino Due

Pin Fisik	Pin Port	Pin Fungsi SAM3X8E
A0	A.16	ADC channel 7
A3	A.22	ADC channel 4
36	C.4	PWM L 1
37	C.5	PWM H 1
38	C.6	PWM L 2
39	C.7	PWM H 2
40	C.8	PWM L 3
41	C.9	PWM H 3

ADC dikonfigurasi standar, dengan resolusi 12 bit dan waktu konversi mengikuti waktu *timer*. Guna menjaga *sample* dan tidak mengganggu maupun terganggu oleh kinerja program lainnya, digunakan *peripheral* yang merupakan fitur ARM, yaitu *Direct Memory Access* (DMA). Pada dasarnya, DMA membaca nilai dari ADC saat satu konversi selesai dan menyalurkannya ke *memory*, dimana seluruh proses tersebut berlangsung independen dari prosesor. Pada program, digunakan *Peripheral DMA Controller* (PDC) yang merupakan salah satu dari dua DMA yang tersedia di SAM3X8E dan dapat digunakan dengan ADC. Jadi, ketika PDC telah terkonfigurasi dan diaktifkan, proses ADC channel 7 dan prosesor utama tidak akan saling mengganggu. Jika transfer DMA selesai, maka data akan tersedia di *memory buffer*.



Gambar 3.6 Timing diagram proses modulasi PWM

PWM juga dihasilkan mengikuti waktu *timer*. PWM menghasilkan gelombang pulsa 40 kHz dengan *duty cycle* berdasarkan nilai pembacaan ADC. *Timer* menghitung mulai dari 0 hingga 1023 (10 bit) setiap 25 μ s. PWM akan menghasilkan output *on* setiap kali *timer* bernilai 0 dan akan menghasilkan output *off* setiap kali hitungan *timer* menyamai nilai pembacaan ADC / 4.

3.2.3 Time delay

Waktu *delay* (t_d) antar transduser yang bersebelahan didasarkan pada input pengguna melalui sebuah potensiometer. Waktu *delay* akan menentukan besar sudut pembelokan suara. Berdasarkan rumusan yang telah dikaji sebelumnya, terdapat dua parameter yang berpengaruh pada batas sudut pengarahannya yaitu panjang gelombang (λ) dan lebar elemen (A). Panjang gelombang frekuensi 40 kHz adalah 8.5 mm sementara lebar transduser yang digunakan ialah 16 mm. Dari definisi tersebut, bisa didapatkan bahwa sudut pembelokan maksimum dari rancangan ini ialah:

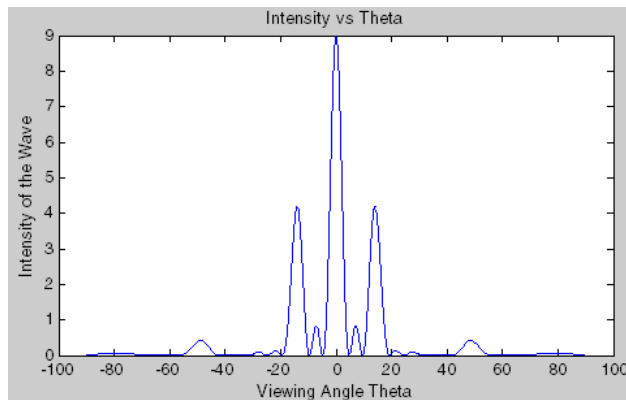
$$\theta_{max} = \sin^{-1} \left(0.514 \cdot \frac{\lambda}{A} \right)$$

$$\theta_{max} = \sin^{-1} \left(0.514 \cdot \frac{8,5 \text{ mm}}{16 \text{ mm}} \right)$$

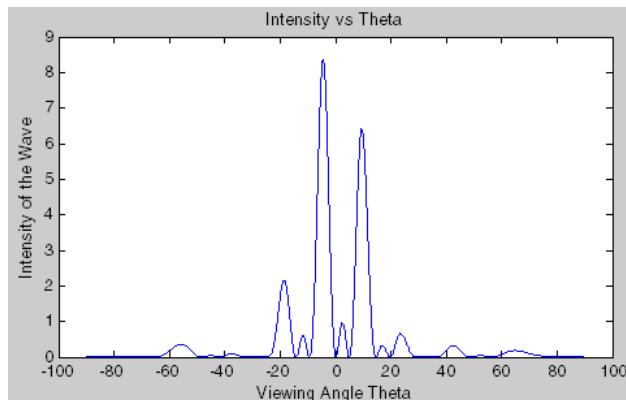
$$\theta_{max} = 15,85^\circ$$

Untuk mengantisipasi bagaimana perilaku sudut pembelokan gelombang, dilakukan simulasi menggunakan aplikasi MATLAB. *Code* yang digunakan untuk simulasi terlampir. Hasil simulasi ditunjukkan pada Gambar 3.7 dan Gambar 3.8 menunjukkan pergeseran *lobe* gelombang konstan 40 kHz dengan mengubah waktu *delay*. Plot atas menunjukkan simulasi tanpa *delay*, sedangkan plot bawah menunjukkan simulasi dengan *delay* sebesar 25 μ s. Dari hasil simulasi juga didapatkan kesesuaian dengan perhitungan teori sebelumnya yang hasilnya menyatakan bahwa sudut pembelokan maksimum berada pada sekitar $\pm 15^\circ$ ke arah kedua sisi. Jika waktu *delay* terus dinaikkan, maka *main lobe* perlahan akan kehilangan daya sedangkan *side lobe* perlahan semakin kuat hingga melewati intensitas *lobe* sebelumnya.

Dari simulasi dapat ditunjukkan bahwa *main lobe* bisa diatur hanya dengan memanfaatkan waktu *delay*. Hal ini mengindikasikan tidak diperlukan pengolahan sinyal tambahan – semisal FFT (*Fast Fourier Transform*) untuk mengatur fase di domain frekuensi – untuk bisa menggeser sinyal secara merata, sehingga proses perhitungan cukup sederhana dan bisa dilakukan secara cepat oleh mikrokontroler tanpa mengganggu proses set data selanjutnya.



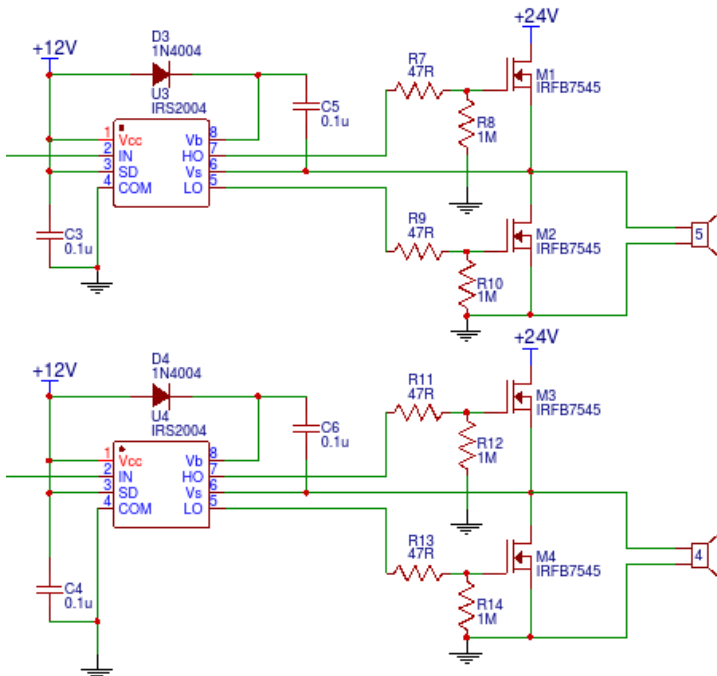
Gambar 3.7 Simulasi *beam pattern* tanpa *delay*



Gambar 3.8 Simulasi *beam pattern* dengan *delay*

3.2.4 Amplifier

Dari output Arduino Due, dihubungkan terlebih dahulu dengan sebuah *opto-coupler* untuk menghindari kontak langsung antara Arduino Due dengan tegangan kerja *half-bridge driver* senilai 12V, sehingga menghindari potensi merusak *board* Arduino Due akibat kelebihan tegangan. H-Bridge di-drive oleh IRS2111. IC tersebut berisi logika untuk bergantian antara sisi *high* dan *low* dari *half-bridge*, memastikan agar suplai tegangan dan *ground* tidak pernah terhubung langsung. Selain itu, digunakan dioda dan kapasitor tambahan untuk menghasilkan tegangan yang dibutuhkan bagi MOSFET. *Opto-coupler* dan *half-bridge driver* tidak dapat menggunakan tegangan suplai 24V secara langsung. H11L1 memiliki batas tegangan maximum 16V sementara IRS2111 memiliki kebutuhan minimum 10V. Untuk itu, digunakan LM7812.



Gambar 3.9 Skematik *amplifier* kelas D

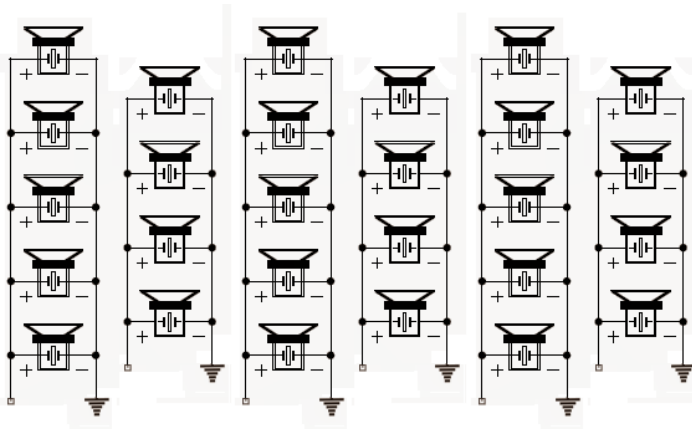


Gambar 3.10 Realisasi rangkaian *amplifier*

3.2.5 *Parametric Array*

Transduser ultrasonik yang digunakan serupa dengan transduser ultrasonik yang lazim dijumpai pada modul sensor jarak. Transduser piezoelektrik yang digunakan memiliki beberapa karakteristik. *Bandwidth* frekuensi nominal dari transduser adalah sebesar 40 kHz dengan simpangan hingga ± 1.0 kHz. Transduser tersebut memiliki diameter 16mm dan tinggi 12mm. Nilai kapasitansi transduser adalah sebesar 2100 pF. Level tekanan suara adalah sekitar 115 dB, diukur pada frekuensi 40 kHz, jarak 30 cm, dan tegangan 10 Vrms. Jarak di mana level suara dapat bertahan sekitar 100 dB adalah hingga 70 cm di depan transduser, dan terus melemah untuk jarak yang semakin jauh.

Sejumlah transduser ultrasonik tersebut – tepatnya 27 unit – dirangkai seperti pada Gambar 3.11 dan disusun sedemikian rupa pada PCB menyerupai matriks, masing-masing berjarak 1 mm antara satu dan lainnya, membentuk *parametric array* akustik. Transduser-transduser ultrasonik dihubungkan secara paralel, terbagi menjadi 6 *channel* berbeda dengan masing-masing sumber gelombang sendiri. Hasil realisasi tampak pada Gambar 3.12.



Gambar 3.11 Skematik rangkaian *parametric array*



Gambar 3.12 Realisasi *parametric array*

BAB IV

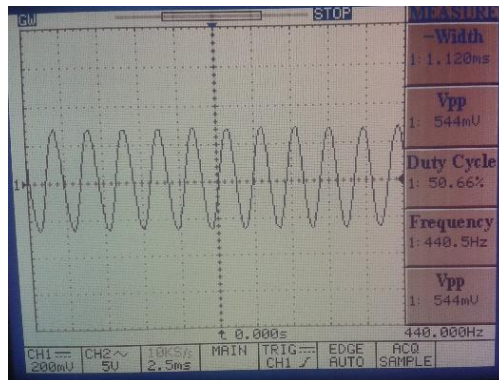
PENGUJIAN DAN ANALISIS

4.1 Analisis elektrik

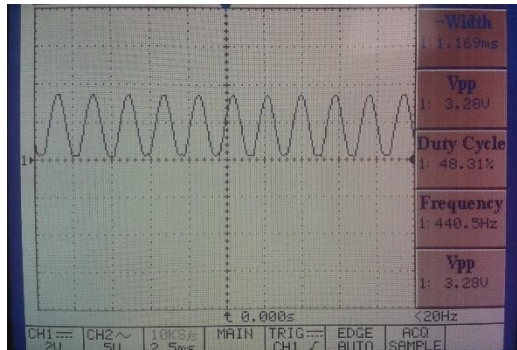
Keseluruhan sistem memiliki beberapa tahapan proses mulai dari input sinyal audio hingga output akhir sinyal PWM 40 kHz termodulasi dengan variasi *duty cycle*. Dilakukan pengujian terhadap setiap tahapan subsistem untuk mengetahui perilaku sistem.

Pertama, dilakukan pengukuran sinyal audio menggunakan osiloskop. Sinyal audio berasal dari perangkat pemutar musik eksternal. Hasil osiloskop terlihat pada Gambar 4.1. Tampak bahwa gelombang sinus audio terbaca memiliki frekuensi 440,5 Hz dengan V_{pp} 544 mV.

Selanjutnya, pada bagian *preamp*, sinyal audio input berhasil diberi *offset bias* dan diamplifikasi ke rentang 0–3,3V bagi Arduino Due. Sinyal hasil *preamp* terbilang cukup baik. Hasilnya terlihat pada Gambar 4.2. Tampak pada osiloskop bahwa sinyal memiliki *bias* menjadi sinyal DC dengan tegangan *peak* 3,28 V dan tetap mempertahankan frekuensi sebesar 440,5 Hz.



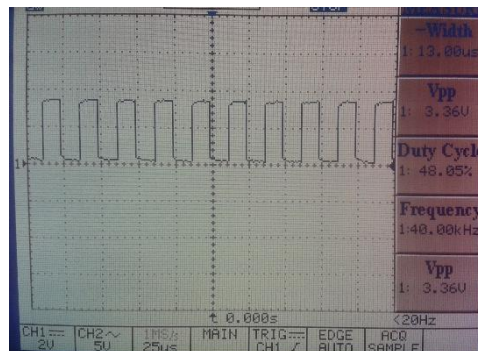
Gambar 4.1 Sinyal audio 440 Hz



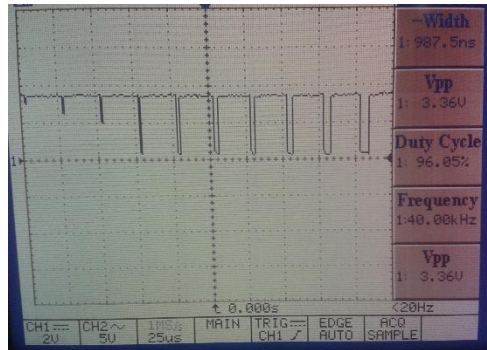
Gambar 4.2 Sinyal output *pre amp*

Board Arduino Due, diinisiasi untuk menghasilkan gelombang PWM berfrekuensi 40 kHz dengan *duty cycle* 50%. Sinyal ini terlihat pada Gambar 4.3. Tampak bahwa Arduino dapat menghasilkan gelombang persegi DC dengan tegangan *peak* 3,36V. Frekuensi yang dihasilkan tepat 40 kHz dengan *duty cycle* awal 48,05%.

Arduino Due diprogram untuk melakukan proses sampling ADC untuk modulasi PWM dengan frekuensi 40 kHz. Hasil modulasi pada Gambar 4.4 memperlihatkan gelombang persegi dengan *duty cycle* perlahan mengecil. Tampak bahwa gelombang PWM memiliki *duty cycle* yang dinamis sebagai efek modulasi. Tegangan *peak* tetap bernilai 3,36V.



Gambar 4.3 Sinyal Arduino PWM 40 kHz, *duty cycle* 50%



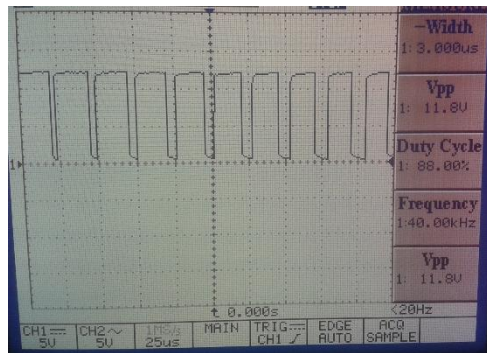
Gambar 4.4 Sinyal Arduino PWM termulasi, *duty cycle* dinamis

Selain itu, Arduino Due juga diprogram untuk melakukan *time delay* berdasarkan input potensiometer dari pengguna. Hasil perbandingan dua gelombang dengan *time delay* dapat dilihat pada Gambar 4.5. Terlihat bahwa terdapat perbedaan fase antara dua gelombang PWM yang dihasilkan.

Sinyal dari Arduino masing-masing diumpankan ke *opto-coupler* H11L1. Pada bagian internal, *opto-coupler* memiliki sebuah transistor *open-collector*. Meskipun bertujuan untuk mengamankan *board* Arduino, *opto-coupler* secara efektif menguatkan sinyal hingga 12V. Output dari *opto-coupler* terlihat pada Gambar 4.6. Dari osiloskop terlihat V_{pp} senilai 11,8V dengan frekuensi tetap terjaga 40 kHz.



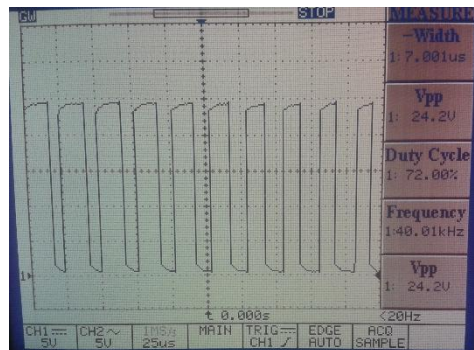
Gambar 4.5 Dua sinyal PWM dengan *delay*



Gambar 4.6 Sinyal output *opto-coupler* H11L1

Pada sisi output menuju transduser ultrasonik, nilai tegangan puncak dapat mencapai 24V. Hasil penguatan akhir ini tampak pada Gambar 4.7. Terlihat bahwa tegangan PWM tetap berfrekuensi 40 kHz namun dengan V_{pp} senilai 24,2 V.

Pada titik ini, secara keseluruhan input audio telah berhasil dimodulasi ke gelombang persegi 3,3V untuk kemudian berhasil dikuatkan hingga 24V.



Gambar 4.7 Sinyal output *amplifier* kelas D

4.2 Analisis akustik

Pengukuran fokus arah *main lobe* dilakukan dengan pengukuran intensitas suara menggunakan sebuah *sound level meter* di area sekitar depan *speaker*. Pertama dilakukan pengukuran level suara *ambient* di lingkungan pengukuran, yaitu didapatkan nilai 59,0 dB. Kemudian diinputkan suara audio ke sistem berupa frekuensi sinusoidal konstan 440 Hz. Level suara di daerah depan *speaker* diukur pada beberapa sudut berbeda dengan selisih 15°. Pengukuran dilakukan pada dua jarak berbeda, yaitu pada jarak 30 cm dan jarak 1 meter dari *speaker*.

Nilai yang terbaca pada *sound level meter* cukup fluktuatif. Oleh sebab itu, dilakukan pencatatan data sebanyak tiga kali dari setiap posisi sudut untuk kemudian dihitung rata-ratanya. Hasil rata-rata pengukuran ditunjukkan pada Tabel 4.1 dan Tabel 4.2. Dari hasil tersebut, terlihat bahwa fokus *main lobe* cukup terarah di area depan *speaker*, nyaris tanpa *side lobe*. Selain itu, tampak perbedaan level suara antara dua jarak yang berbeda. Suara pada jarak 30 cm memiliki level intensitas yang lebih tinggi dibandingkan pada jarak 1 meter. Ini mengindikasikan pelemahan level suara yang relatif signifikan seiring pertambahan jarak.



Gambar 4.8 Kegiatan pengukuran
(kiri) pada jarak 30 cm dan (kanan) pada jarak 1 meter.

Tabel 4.1 Pengukuran level suara tanpa *delay* pada jarak 30 cm.

Sudut	dB
-75°	59,6
-60°	61,8
-45°	65,2
-30°	68,0
-15°	70,2
0°	72,6
15°	70,5
30°	68,5
45°	65,4
60°	62,1
75°	59,9

Tabel 4.2 Pengukuran level suara tanpa *delay* pada jarak 1 meter

Sudut	dB
-75°	59,3
-60°	61,5
-45°	63,7
-30°	66,2
-15°	68,1
0°	69,4
15°	67,8
30°	66,1
45°	63,3
60°	61,9
75°	59,2

Perihal akurasi reproduksi suara frekuensi konstan, dilakukan pengukuran dengan membandingkan beberapa macam frekuensi konstan antara input dan hasil output, serta dilakukan pembandingan subyektif oleh pendengar akan kesamaan nada yang didengar. Macam-macam frekuensi audio yang diuji adalah 250 Hz, 440 Hz, 1 kHz, dan 10 kHz. Pada hasil pengukuran frekuensi output didapatkan frekuensi utama yang sama dengan frekuensi input, namun terdapat *noise* serta harmonisa yang

tercampur dengan frekuensi tersebut. Dari hasil pengukuran subyektif dapat dikonfirmasi bahwa nada yang didengar adalah sama meskipun dengan intensitas volume output yang lebih rendah.

Sementara itu, untuk reproduksi musik atau lagu, terdapat distorsi dan *noise* yang cukup signifikan. Secara subyektif, pendengar masih dapat mengenali lagu yang diputar, meskipun dengan kualitas yang cukup rendah. Hal ini disebabkan oleh beberapa faktor, seperti resolusi *sampling* serta resolusi PWM yang tidak mencapai standar audio musik 16 bit, *noise* hasil interferensi gelombang ultrasonik, dan lain sebagainya.

Untuk perihal pengarahannya menggunakan *time delay*, dilakukan juga perbandingan berdasarkan pengukuran obyektif dan pendengaran subyektif. Sayangnya, hasil pembelokan 15° tidak efektif terukur pada *sound level meter* serta tidak dapat dirasakan oleh pendengar.

Tabel 4.3 Pengukuran level suara dengan *delay* 25 μ s pada jarak 30 cm

Sudut	dB
-75°	59,5
-60°	62,1
-45°	65,2
-30°	66,9
-15°	69,4
0°	70,3
15°	69,3
30°	67,1
45°	65,4
60°	62,0
75°	59,2

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil keseluruhan proses perancangan dan uji coba, didapatkan beberapa kesimpulan. Pertama, modulasi frekuensi audio ke frekuensi ultrasonik 40 kHz dapat dilakukan dengan metode modulasi PWM, yaitu dengan memvariasi *duty cycle* gelombang PWM ultrasonik berdasarkan amplitudo gelombang audio. Proses tersebut juga mengamplifikasi V_{pp} input sinyal audio sebesar 500 mV hingga menjadi 24 V sebagai V_{pp} output. Kemudian, gelombang ultrasonik dengan variasi *duty cycle* dapat mengalami demodulasi di udara kembali menjadi frekuensi audio dan dapat didengar oleh manusia.

Pemfokusan gelombang suara audio dengan menggunakan *parametric array* yang memancarkan gelombang ultrasonik termodulasi memiliki fokus *main lobe* yang sangat baik meliputi daerah depan *speaker*, yaitu dengan lebar sudut 30° ke arah kanan dan kiri. Reproduksi suara audio frekuensi konstan cukup baik meskipun dengan level suara yang lebih rendah yaitu hanya berkisar total 10 dB. Selain itu, reproduksi suara musik tidak optimal sebagai efek dari resolusi *sampling* yang tidak terlalu tinggi. Keseluruhan reproduksi audio memiliki sedikit *noise* akibat hasil interferensi gelombang ultrasonik.

Sementara itu, pengarahan gelombang suara ke sudut tertentu belum cukup efektif. Hal ini sebagai efek keterbatasan karakteristik rancangan, antara lain rasio panjang dan lebar gelombang serta posisi kelengkungan elemen transduser.

5.2 Saran

Adapun beberapa saran yang didapat diberikan untuk penelitian topik serupa dan pengembangan selanjutnya, antara lain: menambahkan *knob* volume; penggunaan resolusi digital yang lebih besar untuk reproduksi suara yang lebih dinamis; melakukan proses pengolahan sinyal digital lanjutan dengan mikrokontroler; diperlukan pemakaian elemen transduser dengan lebar yang lebih kecil serta kemungkinan penyusunan realisasi *parametric array* yang lebih melengkung agar pengarahan gelombang suara dapat lebih efektif dan melebar.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] F. J. Pompei, “*Sound From Ultrasound: The Parametric Array as an Audible Sound Source*,” Massachusetts Institute of Technology, Cambridge, 2002.
- [2] C. Woodford, “*Directional Loudspeakers*,” Bloomsbury Publishing, 2009.
- [3] B. H. Suits, “*Frequencies of Musical Notes for $A_4 = 440$ Hz*,” Department of Physics, Michigan Technological University, [Online]. <http://www.phy.mtu.edu/~suits/notefreqs.html>. [Diakses 30 November 2016].
- [4] A. Green, “*A Very Short History of Ultrasonic Directional Speakers*,” 06 Januari 2016. [Online]. <https://hackaday.io/project/9085-ultrasonic-directional-speaker-v1/log/29967-a-very-short-history-of-ultrasonic-directional-speakers>. [Diakses 30 November 2016].
- [5] W. Ockenga, “*Phase Contrast*,” Leica Science Lab, 09 Juni 2011. <http://www.leica-microsystems.com/science-lab/phase-contrast/>. [Diakses 30 September 2016].
- [6] S. Trenchard dan A. Coppens, “*Experimental study of a saturated parametric array in air*,” Acoustical Society of America, 1980.
- [7] C. Shi, Y. Kajikawa dan W. S. Gan, “*An overview of directivity control methods of the parametric array loudspeaker*,” Cambridge University, 2014.
- [8] C. A. Balanis, dalam *Antenna Theory: Analysis and Design*, John Wiley & Sons, 2015, p. 302–303.
- [9] E. Szoka dan T. Jackson, “*ECE 4670 Final Project: Phased Array Speaker System*,” Cornell University, School of Electrical and Computer Engineering, 01 Mei 2012. http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2012/tcj26_ecs227/tcj26_ecs227/. [Diakses 11 November 2016].
- [10] M. Acebes, D. Florez, N. Thorpe dan T. Aguado, “*Optimization of Phased-Array Transducers for Ultrasonic Inspection in*,” Madrid, 2012.
- [11] K. Miura, “*Ultrasonic Directive Speaker*,” *Elektor Magazine*, no. 03/2011, p. 56, Maret 2011.

- [12] B. Putzeys, A. Veltman, P. van der Hulst dan R. Groenenberg, “All amplifiers are analogue, but some amplifiers are more analogue than others,” dalam *120th Audio Engineering Society Convention*, Paris, 2006.

LAMPIRAN

- Pemrograman untuk Arduino Due dengan Atmel Studio

main.c

```
#include <asf.h>
#include <string.h>
#include "log.h"
#include "init.h"

static uint32_t ticks = 0;
static uint16_t min_adc = 0;
static uint16_t max_adc = 0;
static uint16_t min_dac = 0;
static uint16_t max_dac = 0;

static void loop(void) {
    if (ticks % 10 == 0) {
        max_adc = max_dac = 0;
        min_adc = min_dac = 0xffff;
    }
    if (ticks % 10 == 0) {
        pio_toggle_pin(PIO_PB27_IDX);
    }
}

void TC0_Handler(void) {
    tc_get_status(TC0, 0);
    uint16_t ch0 = (uint16_t)
adc_get_channel_value(ADC, ADC_CHANNEL_7);
    if (ch0 < min_adc) {
        min_adc = ch0;
    }
    if (ch0 > max_adc) {
        max_adc = ch0;
    }

    int32_t duty = ((int32_t) ch0) * US_PERIOD / adc_limit;
    if (duty >= US_PERIOD) {
        duty = US_PERIOD - 1;
    }
}
```

```

    }
    PWM->PWM_CH_NUM[2].PWM_CDTYUPD = duty;
    PWM->PWM_CH_NUM[3].PWM_CDTYUPD = duty;
}

void TC1_Handler(void) {
    tc_get_status(TC0, 1);
    ticks++;
    loop();
}

```

init.h

```
#ifndef INIT_H_
#define INIT_H_
#define US_PERIOD 2100
void init(void);
#endif
```

init.c

```
#include <asf.h>
#include "init.h"
#include "log.h"

static void configure_io_pins(void) {
    pmc_enable_periph_clk(ID_PIOA);
    pio_configure_pin(PIO_PA16_IDX, PIO_INPUT);

    pmc_enable_periph_clk(ID_PIOB);
    pio_configure_pin(PIO_PB15_IDX, PIO_PERIPH_B);
    pio_configure_pin(PIO_PB27_IDX, PIO_OUTPUT_1);

    pmc_enable_periph_clk(ID_PIOC);
    pio_configure_pin(PIO_PC6_IDX, PIO_PERIPH_B);
    pio_configure_pin(PIO_PC7_IDX, PIO_PERIPH_B);

    pmc_enable_periph_clk(ID_PIOD);
    pio_configure_pin(PIO_PC8_IDX, PIO_PERIPH_B);
    pio_configure_pin(PIO_PC9_IDX, PIO_PERIPH_B);
}

static void configure_pwm(void) {
    pwm_channel_disable(PWM, PWM_CHANNEL_2);
    pmc_enable_periph_clk(ID_PWM);
    pwm_channel_t instance2 = {
        .channel = PWM_CHANNEL_2,
        .ul_prescaler = PWM_CMR_CPRE_MCK,
        .alignment = PWM_ALIGN_LEFT,
        .polarity = PWM_LOW,
    };
}
```

```

        .ul_period = US_PERIOD,
        .ul_duty = US_PERIOD / 2,
    };
    pwm_channel_init(PWM, &instance2);
    pwm_channel_enable(PWM, PWM_CHANNEL_2);

    pwm_channel_disable(PWM, PWM_CHANNEL_3);
    pmc_enable_periph_clk(ID_PWM);
    pwm_channel_t instance3 = {
        .channel = PWM_CHANNEL_3,
        .ul_prescaler = PWM_CMR_CPRE_MCK,
        .alignment = PWM_ALIGN_LEFT,
        .polarity = PWM_LOW,
        .ul_period = US_PERIOD,
        .ul_duty = US_PERIOD / 2,
    };
    pwm_channel_init(PWM, &instance3);
    pwm_channel_enable(PWM, PWM_CHANNEL_3);
}

static void configure_timer(void) {
    static uint32_t ul_sysclk;
    ul_sysclk = sysclk_get_cpu_hz();
    uint32_t ul_div, ul_tcclks;

    pmc_enable_periph_clk(ID_TC0);
    tc_find_mck_divisor(40000, ul_sysclk, &ul_div,
    &ul_tcclks, ul_sysclk);
    tc_init(TC0, 0, ul_tcclks | TC_CMR_CPCTRG);
    tc_write_rc(TC0, 0, (ul_sysclk / ul_div) / 40000);
    NVIC_SetPriority((IRQn_Type)ID_TC0, 0);
    NVIC_EnableIRQ((IRQn_Type)ID_TC0);
    tc_enable_interrupt(TC0, 0, TC_IER_CPCS);
    tc_start(TC0, 0);

    pmc_enable_periph_clk(ID_TC1);
    tc_find_mck_divisor(20, ul_sysclk, &ul_div,
    &ul_tcclks, ul_sysclk);
    tc_init(TC0, 1, ul_tcclks | TC_CMR_CPCTRG);
    tc_write_rc(TC0, 1, (ul_sysclk / ul_div) / 20);
    NVIC_SetPriority((IRQn_Type)ID_TC1, 8);
    NVIC_EnableIRQ((IRQn_Type)ID_TC1);
    tc_enable_interrupt(TC0, 1, TC_IER_CPCS);

```

```

        tc_start(TC0, 1);
    }

    static void configure_adc(void) {
        pmc_enable_periph_clk(ID_ADC);
        adc_init(ADC, sysclk_get_cpu_hz(), 5 * 1000 * 1000,
ADC_STARTUP_TIME_10);
        adc_configure_timing(ADC, 4, 0, 1);
        adc_set_bias_current(ADC, 1);
        adc_enable_channel(ADC, ADC_CHANNEL_7);
        adc_configure_trigger(ADC, ADC_TRIG_SW, 1);
        adc_start(ADC);
    }

    void init(void) {
        sysclk_init();
        board_init();
        configure_io_pins();
        configure_pwm();
        configure_timer();
        configure_adc();
    }

```

asf.h

```
#ifndef ASF_H
#define ASF_H

// From module: ADC - Analog-to-digital Converter
#include <adc.h>

// From module: Common SAM compiler driver
#include <compiler.h>
#include <status_codes.h>

// From module: DACC - Digital-to-Analog Converter
#include <dacc.h>

// From module: Delay routines
#include <delay.h>

// From module: FreeRTOS - PDC Interface Layer
#include <freertos_peripheral_control.h>
#include <freertos_peripheral_control_private.h>

// From module: FreeRTOS - UART Interface Layer
#include <freertos_uart_serial.h>

// From module: FreeRTOS mini Real-Time Kernel
#include <FreeRTOS.h>
#include <FreeRTOS_CLI.h>
#include <StackMacros.h>
#include <croutine.h>
#include <list.h>
#include <mpu_wrappers.h>
#include <portable.h>
#include <projdefs.h>
#include <queue.h>
#include <semphr.h>
#include <task.h>
#include <timers.h>

// From module: GPIO - General purpose Input/Output
#include <gpio.h>

// From module: Generic board support
#include <board.h>
```



```

// From module: IOPORT - General purpose I/O service
#include <ioport.h>

// From module: Interrupt management - SAM implementation
#include <interrupt.h>

// From module: PDC - Peripheral DMA Controller Example
#include <pdh.h>

// From module: PIO - Parallel Input/Output Controller
#include <pio.h>

// From module: PMC - Power Management Controller
#include <pmc.h>
#include <sleep.h>

// From module: PWM - Pulse Width Modulation
#include <pwm.h>

// From module: Part identification macros
#include <parts.h>

// From module: SAM3X startup code
#include <exceptions.h>

// From module: System Clock Control - SAM3X/A
implementation
#include <sysclk.h>

// From module: TC - Timer Counter
#include <tc.h>

// From module: UART - Univ. Async Rec/Trans
#include <uart.h>

// From module: USART - Serial interface - SAM
implementation for devices with both UART and USART
#include <serial.h>

// From module: USART - Univ. Syn Async Rec/Trans
#include <usart.h>

// From module: pio_handler support enabled
#include <pio_handler.h>

#endif

```

- Program MATLAB untuk simulasi *phased array* akustik

```
function I = PhasedArrayPlot(N,d,a,lambda,td)
%N untuk banyaknya elemen
%d untuk jarak antar elemen (center to center) (meter)
%a adalah lebar elemen / lebar wavefront (meter)
%lambda adalah panjang gelombang (meter)
%td adalah time delay (detik)

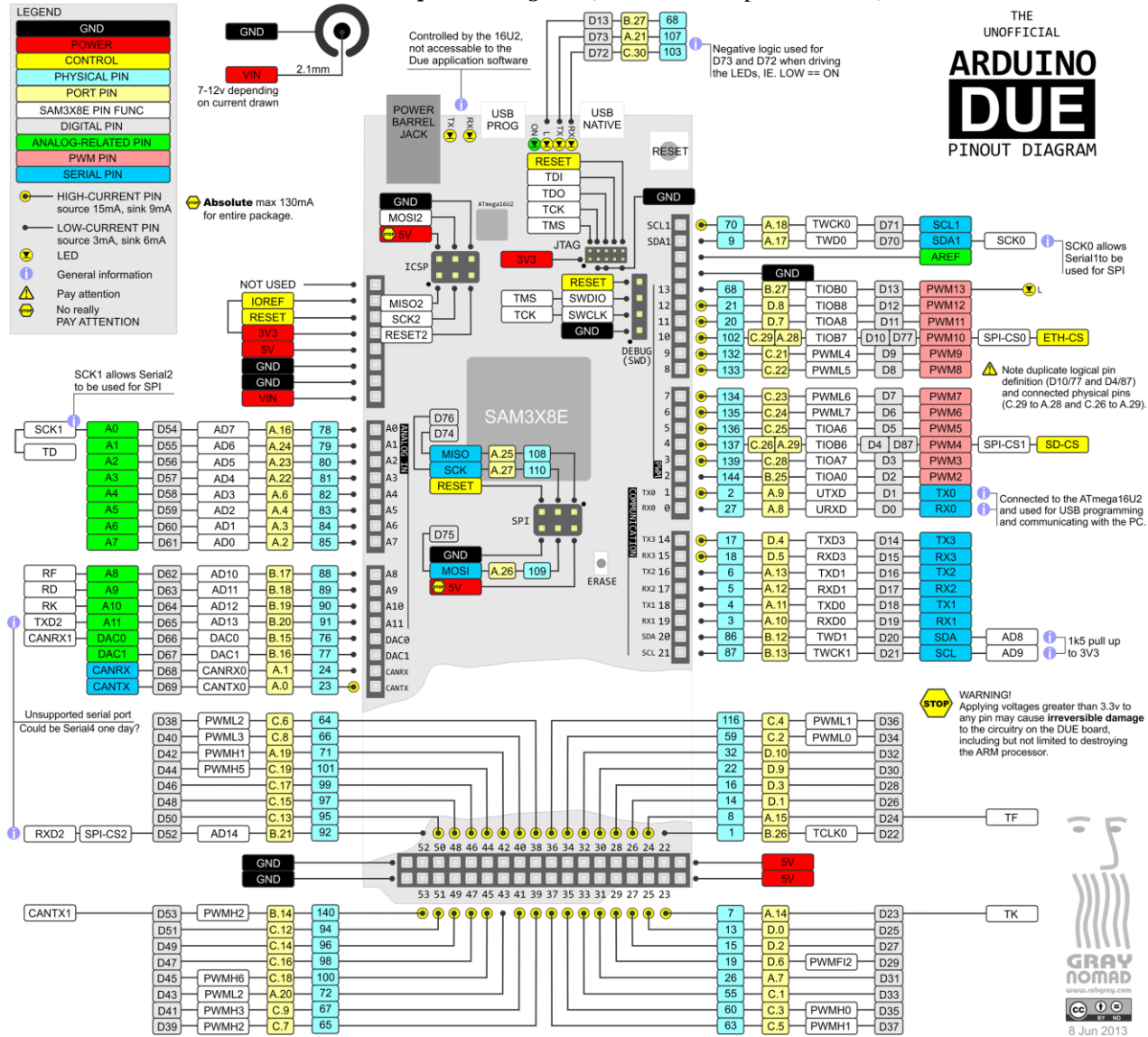
theta = linspace(-90,90,10000);
phi = td/lambda*343.3;

A = sin(pi*a.*sind(theta)/lambda);
B = pi*a.*sind(theta)/lambda;
C = ((2*pi*d.*sind(theta)/lambda)+phi);
D = sin((N/2).*C);
E = sin(C);
I = ((A./B).^2).*((D./E).^2);

figure(1)
plot(theta,I,'b')
xlabel('Viewing Angle Theta')
ylabel('Intensity of the Wave')
title('Intensity vs Theta');

figure(2)
semilogy(theta,I,'b')
xlabel('Viewing Angle Theta')
ylabel('Intensity of the Wave')
title('Intensity vs Theta');
```

Arduino Due pin-out diagram (diambil dari <http://arduino.cc>)



BIODATA PENULIS



Penulis dilahirkan di Surabaya pada tanggal 13 Juni 1995 dengan nama lengkap Domi Lavaró, dari orang tua Roval dan Mira Ratnawati. Penulis menjalani mayoritas masa hidupnya di Surabaya hingga saat ini. Penulis menamatkan pendidikan dasar di SD Darut Taqwa Surabaya (tahun 2006), pendidikan menengah di SMPN 16 Surabaya (tahun 2009) dan SMAN 1 Sidoarjo (tahun 2012). Saat ini penulis berusia 21 tahun dan sedang menempuh studi tingkat Strata 1 di Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Penulis mengambil bidang studi Elektronika sebagai fokus studinya. Selama kuliah penulis juga kerap aktif dalam berbagai kegiatan akademis maupun non-akademis. Akhir kata, penulis dapat dihubungi melalui email: domilavaro@gmail.com.